Graph-based Sequence Clustering through Multiobjective Evolutionary Algorithms for Web Recommender Systems

Gul Nildem Demir
Istanbul Technical University
Computer Engineering Department
Maslak 34469 Istanbul, Turkey
ndemir@itu.edu.tr

A. Sima Uyar
Istanbul Technical University
Computer Engineering Department
Maslak 34469 Istanbul, Turkey
etaner@itu.edu.tr

Sule Oguducu
Istanbul Technical University
Computer Engineering Department
Maslak 34469 Istanbul, Turkey
sgunduz@itu.edu.tr

ABSTRACT

In web recommender systems, clustering is done offline to extract usage patterns and a successful recommendation highly depends on the quality of this clustering solution. In these types of applications, data to be clustered is in the form of user sessions which are sequences of web pages visited by the user. Sequence clustering is one of the important tools to work with this type of data. One way to represent sequence data is through weighted, undirected graphs where each sequence is a vertex and the pairwise similarities between the user sessions are the edges. Through this representation, the problem becomes equivalent to graph partitioning which is NP-complete and is best approached using multiple objectives. Hence it is suitable to use multiobjective evolutionary algorithms (MOEA) to solve it. The main focus of this paper is to determine an effective MOEA to cluster sequence data. Several existing approaches in literature are compared on sample data sets and the most suitable approach is determined.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.5.3 [**Pattern Recognition**]: Clustering—*algorithms*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms

Keywords

graph-based clustering, sequence clustering, multiobjective evolutionary algorithms

GECCO'07, July 7-11, 2007, London, England, United Kingdom

1. INTRODUCTION

Clustering can be defined as finding groups of similar objects in unlabeled data [14, 27]. Data to be clustered usually occurs in two forms: data items given in metric space and data items given as sequences. Most of the research in literature focus on clustering data items of the first type [2, 25]. In the second type of data, there is a similarity or a dissimilarity measure defined between each pair of data items. This kind of data can be found in many real-world application domains, such as biostatistics, medicine, telecommunications, user interface studies, market basket data, and World Wide Web (WWW) page request monitoring. Sequence clustering is one of the important tools to understand and work with this type of data.

One of the ways to represent sequence data is through weighted, undirected graphs. Each sequence in the data sets becomes a vertex of a graph and the pairwise similarities or dissimilarities form the edges connecting the corresponding vertices in the graph. Through this representation approach, the sequence clustering problem becomes equivalent to graph partitioning which is an NP-complete problem. This makes it suitable to use evolutionary algorithms (EA) to solve it. There are some successful studies in literature which use EAs for clustering data items given with pairwise similarity or dissimilarity values. In [22], a multiobjective evolutionary algorithm is used to cluster files. This work does not use a graph-based approach to the problem. In [7] and [8] a graph-based evolutionary approach is used. In [7], there is only one objective and the number of clusters to separate the data into is fixed. However, in [22] and [8], two objectives are used and the cluster count is automatically determined by the EA. As has been shown in [19, 18], [22] and [8], the clustering problem with automatic determination of the number of clusters is better approached with multiple objectives. Results of these studies show that this approach is indeed useful and provides good results.

The experiments performed in this study are part of an ongoing research project on fast and efficient web recommendation systems. All web recommendation systems are composed of two components: an off-line component and an on-line component. Clustering of user sessions is performed in the off-line phase in order to extract usage patterns. The performance of a web recommendation system depends on how well the patterns are extracted from usage data. Therefore, the main focus of this paper is to determine an effective

multiobjective evolutionary algorithm (MOEA) to be used in clustering sequence data. For this purpose, in the following sections, several existing approaches in literature will be empirically compared on sample data sets and the most suitable approach will be determined.

The rest of the paper is organized as follows: Section 2 provides an overview of related work. In section 3, the MOEAs used in this study are explained. Section 4 provides the specifics about the experiments and section 5 provides results and discussions. Section 6 concludes the paper and gives directions for future work.

2. RELATED WORK

An important form of data considered in data mining is sequential data. This kind of data occurs in many application domains, such as biostatistics, medicine, telecommunication, user interface studies, market basket data, and World Wide Web (WWW) page request monitoring. Understanding the structure of such data still remains a challenge. For this reason, sequence clustering has become increasingly important. A representation of sequence data is a weighted, undirected graph G where each sequence in the data set becomes a vertex in the graph. The pairwise similarities of sequences form the edges of the graph. Properties of a graph can then be used to cluster sequences by constructing a set of subgraphs from G. The sequence clustering problem can then be mapped onto a graph partitioning problem.

Most of the graph partitioning algorithms refine the partitions recursively. Such methods generate a hierarchy of clusters presented as a dendogram to the user. At any given point, a set of sequence clusters or subgraphs such as $\{G_1, G_2, ..\}$ are given. The problem in such methods is in deciding whether a cluster, for example G_2, should be split further or not. Different cutoff values may result in different clustering solutions. However, it is very difficult to put a single cutoff value that separates all clusters. Many clustering algorithms use graph properties to cope with this problem. The algorithms in [13, 28] use purely graph theoretic approaches. However, there remains another important issue. The simplest MIN cut algorithm tends to favor partitioning into subgraphs where a subgraph could be very small compared to the others. Various constraints are introduced, such as ratio cut [3], normalized cut [24], and min-max cut [12], etc. to remedy this problem.

As has been shown in [19, 18], if the number of clusters is not given beforehand and the algorithm has to determine them automatically, it is better to use a multiobjective approach. Two successful methods exist in literature which address the problem of clustering data defined through pairwise similarities or dissimilarities using a MOEA technique.

The MultiObjective Clustering with automatic K-determination (MOCK) [19], is a MOEA based on PESA-II [5]. It optimizes two complementary objective functions: *overall deviation* and *connectivity*. The algorithm generates clusterings with different cluster counts and quality. It also contains a final step to select good solutions from the Pareto approximation set and to determine the number of clusters in the data set. MOCK works on data sets with numerical feature vectors. It is adapted to data sets defined using pairwise similarity or dissimilarity information under the name MOCK-around-medoids (MOCK-am) [22].

The approach proposed in [8] is also a MOEA but it combines the objectives using an aggregated sum approach. This requires the determination of appropriate weights for each objective and the weight values that work better vary across different data sets. Using sub-optimal weight vaues decreases performance. So this becomes an optimization problem in itself. It also optimizes two complementary objectives, namely the *min-max-cut* and the *global silhouette index*. It is specifically developed to work with data sets defined using pairwise similarity or dissimilarity information.

3. THE MOEAS FOR GRAPH-BASED CLUSTERING

In previous work by Handl et. al. [19, 18, 20, 21, 23], it has been shown that the data clustering problem is best approached using multiple objectives. When working with multiple objectives, the search process consists of two stages. In the first stage, a set of good solutions needs to be found. In the second stage, which can be called the decision making (DM) stage, a suitable solution for a particular application needs to be selected. This is an important phase of a multiobjective optimization problem. The DM approaches can be classified as:

- apriori preference approaches where the decision of the relative importance of the objectives is made before the search and a decision maker combines the objective functions into a scalar cost function converting the problem into a single objective optimization problem
- progressive preference articulation where DM and optimization are intertwined and partial preference information are provided at different stages of the search
- aposteriori approaches where a decision maker is presented with a set of pareto optimal solutions and the decision maker selects the suitable one based on experience and also the requirements of a specific application

It is usually more preferable in most application domains to use DM approaches of the third type. Due to the fact that EAs are population based search methods where the search goes on simultaneously at different points in the search space and no assumptions need to be made about the shape or the continuity of the pareto front, they are especially suitable for obtaining the pareto front in an optimization problem with multiple objectives. There are several very successful MOEAs in literature among which SPEA2 [32], PESA-II [5] and NSGA-II [10] can be named. Further information on evolutionary algorithms for multiobjective problems can be found in [4].

Several successful EAs have been proposed in literature for the clustering problem. However since the main focus of this paper is specifically on clustering data items given with pairwise similarity or dissimilarity information, only the approaches suitable for this problem will be presented here. As mentioned in the previous section, two successful MOEA approaches for clustering data items of this type exist in literature. These two approaches will be further explained in the following subsections.

3.1 MOCK-am

MOCK-am is an extension of the original MOCK algorithm to be used with data which is given as pairwise similarities or dissimilarities between the items. Like MOCK, MOCK-am is also based on the elitist MOEA, PESA-II. It

tries to optimize two conflicting objectives, namely: *overall deviation* (OD) and *connectivity* (CO).

PESA-II is an elitist MOEA with a hyper-grid crowding strategy in the objective space. It maintains two populations: an internal population (IP) with a constant size and an external population (EP) with a limited size. The EP contains good solutions which form an approximation to the Pareto front. The IP consists of candidate solutions for the external population. The objective space is divided implicitly into hyper-boxes. Every solution in the EP belongs to a certain hyper-box. A non-dominated solution in IP can be moved to EP under two conditions:

1. EP does not exceed its size limit
2. EP is full but the non-dominated candidate will not be placed in the crowded hyper-box. Consequently one solution from the most crowded hyper-box has to be removed from the EP.

In PESA-II, the selection operation depends also on the hyper-grid structure. Firstly, a populated hyper-grid is selected randomly and a random individual is chosen from that hyper-grid. The flow of the algorithm is given in Algorithm 1. Detailed information on PESA-II can be found in [5].

Algorithm 1 PESA-II

```
Set IP and EP to the empty set
Initialize and evaluate IP
Update EP
while max no of generations not reached do
    Selection: select individuals from EP
    Cross-over: create child through heuristic uniform cross-over
    Mutation: mutate child
    Fitness Evaluation: evaluate children
    Update EP according to the crowding strategy
end while
Return EP
```

In MOCK-am, individuals are encoded according to the locus-based adjacency representation. Each individual g contains N genes, where N is the size of the data set. Gene values vary in the range [1,N]. The value j of the ith gene corresponds to a link between the items i and j which indicates that they are in the same cluster. The initialization is based on minimum spanning trees (MST). For a given data set the MST is generated using Prim's algorithm. The first individual becomes the complete MST and the ith individual is created by removing the i largest links from the MST. This encoding enables the use of a standard uniform cross-over operator. The mutation operator, *restricted nearest neighbour mutation*, randomly links an item only to one of its L nearest neighbours.

The first objective function *overall deviation* OD, calculated as in Eq. 1, sums the distances of items to their cluster medoid. The distance function is the dissimilarity between the item and the medoid of its cluster.

$$OD(C) = \sum_{C_k \in C} \sum_{i \in C_k} \delta(i, \mu_k) \tag{1}$$

where C is the set of all clusters and μ_k is the medoid of the cluster C_k

The second objective *connectivity* (CO), calculated as in Eq. 2, checks whether neighbouring items are in the same cluster.

$$CO(C) = \sum_{i=1}^{N} \sum_{j=1}^{L} x_{i,nn_i(j)} \tag{2}$$

where $x_{r,s} = \frac{1}{j} if \nexists C_k : r, s \in C_k$, 0 otherwise. In this formula, N is the vertex count, L is a parameter for nearest neighbour count, $nn_i(j)$ is j-th nearest neighbour of vertex i.

MOCK-am includes a further step, namely the automatic solution selection. However this step is not implemented in this study and the algorithm returns the whole EP instead of suggesting an appropriate solution from the EP. Further details on MOCK can be found in [19, 18] and on MOCK-am in [22].

3.2 GraSC

The Graph-based Sequence Clustering (GraSC) algorithm is an extension of the approach proposed in [8]. It has been modified to use a MOEA to handle multiple objectives instead of an aggregated sum technique. GraSC uses SPEA2 which also is an elitist MOEA. It tries to optimize two objectives, namely: *min-max cut* and *the silhouette index*. SPEA2 assigns fitness values to individuals according to dominance and density criteria. The selection of individuals is also based on these criteria. Similar to PESA-II, it maintains two populations: a current population and an archive population. In each generation, the non-dominated solutions in the current population and in the archive are copied into the archive of the next generation. Selection occurs from both the population and the archive. The algorithmic flow of SPEA2 is given in Algorithm 2. Detailed information on SPEA2 can be found in [32].

Algorithm 2 SPEA2

```
Randomly initialize population P_0 and create empty archive population P̄_0
while while max no of generations not reached do
    Fitness assignment: calculate fitness values of individuals in P_t and P̄_t
    Environmental selection: copy nondominated individuals in P_t and P̄_t to P̄_{t+1}
    Mate selection: select parents from P̄_{t+1} based on binary tournement selection
    Cross-over: create child through heuristic uniform cross-over
    Fitness Evaluation: evaluate children
    Place children in P_{t+1}
end while
Return nondominated individuals in P̄_{t+1}
```

In the fitness assignment step, a strength value $S(i)$, a raw fitness $R(i)$ and a density value $D(i)$ are calculated for each individual i in the population P_t and the archive $\bar{P}_t$. $S(i)$ shows the number of dominated solutions by the ith individual. $R(i)$ is the sum of strength values of its dominators. The density $D(i)$ is determined as follows: For each individual i, its distance to individuals is calculated and the

distances are sorted in increasing order. The kth item σ_i^k where k is usually chosen as the square root of whole population size, is used for $D(i)$ calculation: $D(i) = \frac{1}{\sigma_i^k + 2}$

The sum of $R(i)$ and $D(i)$ gives the final fitness $F(i)$ of individual i. In the environmental selection step, the nondominated individuals, whose fitness is lower than 1 are placed in $\bar{P}_{t+1}$. If the number of such individuals is bigger than the archive size, the archive is truncated with a special operator. If the nondominated individuals cannot fill the archive exactly, best dominated individuals are copied to $\bar{P}_{t+1}$. As mentioned above, the parents can be selected from both the current population and the archive based on their fitness value $F(i)$ using tournament selection. The resulting child is placed into the population of the next generation after recombination and variation.

In GraSC, the representation and genetic operators are the same as proposed in [8]. Each individual g contains N genes where N is the total number of vertices in the graph. Each gene corresponds to a vertex and the value of the gene denotes the cluster number the vertex is placed in. However, for the same k-clustering of a graph there can be $k!$ possible genotypes if the cluster numbers lie in [0, k-1]. To overcome this problem, a post-processing step is added after the initialization of individuals and operators. In this step the individual is processed from left to right and the clusters are re-numbered in increasing order. Individuals are initialized randomly: Each gene is assigned a random value between 0 and $maxCluster$ where the $maxCluster$ parameter is the maximum allowed number of clusters in a partitioning. For each iteration, N children are generated. Each child is created from two parents selected through binary tournament selection. The heuristic cross-over operator operates as follows: Initially all vertices in the graph are regarded as uncovered. At each step, one of the uncovered vertices and one of the parents is selected randomly. The uncovered vertices in the cluster which contains the selected vertex in the selected parent are grouped into one cluster and the newly covered vertices are marked as covered. This process continues until all vertices are covered. At the end, the child contains partial clustering information from both of its parents. As noted in [8], the cross-over operator tends to increase the number of clusters. To remedy this, in [8], the heuristic disband operator is applied to the child after crossover. In this step, the vertices belonging to the cluster with the minimum intra-cluster similarity are placed in the closest cluster. The closest cluster is defined as the cluster with the maximum average similarity to the vertex. A standard mutation operator is used. The cluster number of each vertex is replaced by a new number in the given interval based on a given mutation probability. After mutation, some clusters may become empty. The cluster numbers are enumerated again as a post-processing step. In GraSC, a constraint on the minimum number of vertices in a cluster ($minNode$) is defined. After the completion of the application of the genetic operators, if there are any clusters with less vertices than $minNode$, each vertex in that cluster is moved to the closest cluster.

The first objective, the *min-max-cut* (MMC) [11] function, calculated as in Eq. 3 aims to maximize the similarity within each subgraph while trying to minimize the similarity between the subgraphs.

$$MMC(G) = \sum_{m=1}^{k} \frac{cut(G_m, G \setminus G_m)}{\sum_{v_i v_j \in G_m} E(v_i, v_j)} \tag{3}$$

In this formula, $cut(G_m, G \setminus G_m)$ is the sum of edge weights between the vertices in G_m and in the rest of the graph $G \setminus G_m$. $E(v_i, v_j)$ gives the weight of the edge between the nodes v_i and v_j. The edge weights correspond to the pairwise similarity values between data items. The MMC function should be minimized. For the minimum MMC value 0, all vertices will be placed in the same cluster. Namely this objective tends to decrease the number of clusters down to one where all vertices are in the same cluster. The second objective, *the global silhouette value* (GS) [30] calculated as in Eq. 4, Eq. 5 and finally Eq. 6 is chosen to balance the MMC function.

$$s(v_i) = \frac{b_i - a_i}{max(b_i - a_i)} \tag{4}$$

where n_j gives the number of vertices in cluster C_j, a_i is the average dissimilarity between v_i and other vertices in C_j, b_i is the minimum average dissimilarity between v_i and other clusters. In this work the dissimilarity values are computed as $1 - E(v_i, v_j)$. For each cluster C_j a silhouette index S_j is assigned as in Eq. 5.

$$S_j = \frac{\sum_{i=1}^{n_j} s(v_i)}{n_j} \tag{5}$$

The final formula of GS is as in Eq. 6.

$$GS = \frac{\sum_{j=1}^{k} S_j}{k} \tag{6}$$

GS has its highest value if all clusters are composed of only one vertex. Thus it acts in just the opposite direction of MMC. GS is a cluster validation index and can be used to compare the qualities of clusterings with different number of clusters. It takes on values in the range [-1,1]. For a good clustering, it gets closer to 1 and to -1 otherwise. GS should be maximized. To apply the MOEA, both of the objectives should be taken as maximization. So the MMC value is converted to $1/(1 + MMC)$ throughout the rest of the paper.

Details on the representation, operators and the implementation of the objectives used in GraSC can be found in [8].

4. EXPERIMENTS

The aim of the experiments is to determine an effective MOEA to be used on data defined through pairwise similarity or dissimilarity information. For the purposes of the experiments, GraSC and MOCK-am variations will be tested on two sets of test data. The first data set (CNET) is from Clark Net web server which is a full internet access provider for the Metro Baltimore-Washington DC area [1]. Firstly, user sessions are extracted from the data set according to [6]. Then, since each session consists of a sequence of URL requests, the pairwise similarities of these user sessions are calculated with an algorithm based on FastLSA [26]. Each user session corresponds to a vertex on a graph and these session similarities give the edge weights between the corresponding vertices. The resulting graph consists of 4792

vertices and 444884 edges. The second data set (FILES) is obtained as a dissimilarity matrix from [17]. The entries in the matrix give information about pairwise dissimilarities of some computer files based on Universal Similarity Metric (USM) [29]. The matrix contains 911 rows and 911 columns.

MOCK-am and GraSC use different initialization methods, MOEAs, operators and objectives. To see the individual effects of the selected MOEAs, the objective functions and other algorithmic details like representation of candidate solutions, initialization and genetic operators, different variations of MOCK-am and GraSC which contain various combinations of the above are implemented. All implemented variations are given in Table 1. By comparing the different variations with each other it will be possible to identify the best initialization method, operator group and MOEA combination.

For GraSC, default algorithmic details (*gs-default*) denotes direct encoding (DE), random initialization (RI) and the default operator group consists of the heuristic crossover operator, standard mutation and the heuristic disband operator. Default algorithmic details for MOCK-am (*mock-default*) are locus-based adjacency representation (LAR), initialization based on minimum spanning tree (MST) and the default MOCK operators which consist of uniform crossover and restricted nearest neighbour mutation.

Table 1: Implemented algorithm variations

Variations	MOEA	Obj.	Algorithmic Details
A1	SPEA2	MMC,GS	gs-default
A2	PESA-II	MMC,GS	gs-default
B1	SPEA2	MMC,GS	DE,MST,def.op.
B2	PESA-II	MMC,GS	DE,MST,def.op.
C1	SPEA2	MMC,GS	mock-default
C2	PESA-II	MMC,GS	mock-default
D1	SPEA2	MMC,GS	LAR,RI,def.Mock op.
D2	PESA-II	MMC,GS	LAR,RI,def.Mock op.
E1	SPEA2	OD,CO	gs-default
E2	PESA-II	OD,CO	gs-default
F1	SPEA2	OD,CO	mock-default
F2	PESA-II	OD,CO	mock-default

The general parameter settings for GraSC and MOCK-am used in the experiments are given in Table 2. The L value of the restricted nearest neighbour mutation and medoid computation is selected as 10. For PESA-II the EP size is 1000 and the IP size is 10. The resolution of hypergrid per dimension is 10. For SPEA-II the population size is 50 and the archive size is 20. The general MOCK-am settings are selected as suggested in [22].

Table 2: General Settings

Parameter	GraSC	MOCK-am
Num.of Gen.	1000	1000
Recom. Rate	1	0.7
Mutation Rate	1/N	1/N
MinNode	2	2

Each variation is run 10 times. For each specific variation, the quality of its runs are calculated by *dominance ranking*: For each approximation set, the number of sets by which the set is dominated is counted [15]. The best and the worst runs and the average rank are identified according to this dominance ranking. The approximation set with an average dominance rank is selected for further evaluation of the corresponding variation.

To compare the approximation sets of the different variations the *binary ϵ-indicator* [33] is used. The general definition of binary ϵ-dominance is given as follows: In a minimization problem with n positive objectives an objective vector $z^1 = z_1^1, z_2^1, \ldots, z_n^1$ ϵ-dominates another objective vector $z^2 = z_1^2, z_2^2, \ldots, z_n^2 (z^1 \succeq_\epsilon z^2)$ if $\forall 1 \leq i \leq n : z_i^1 \leq \epsilon z_i^2$ for a $\epsilon > 0$

The binary ϵ-indicator I_ϵ is defined as in Eq. 7.

$$I_\epsilon = inf\{\forall z^2 \in B \exists z^1 \in A : z^1 \succeq_\epsilon z^2\} \tag{7}$$

where A and B are two approximation sets. The boolean function $F := (I_\epsilon(A,B) \leq 1 \wedge I_\epsilon(B,A) > 1)$ is a comparison method to show the relation between A and B. If F returns *true*, A is a better approximation than B. Otherwise it is not possible to determine which one is better. This indicator is used in this study to compare different variations using the same set of objective functions.

In the final stage, the best variation for each objective group is identified. The pareto front for each is plotted and the most promising solutions are manually selected from the corresponding graphs by an expert on web mining. Several cluster validation indices [16] exist in literature to assess the quality of clustering solutions. Since the approaches tested in this study allow the determination of clusters automatically, the selected cluster validity index should be able to compare clustering solutions having a different number of clusters. Two validity indices are selected as suitable for this purpose: namely the silhouette index [30] and the Davies-Bouldin index [9]. The silhouette index is an objective in GraSC, therefore a second index is also chosen to provide a fair comparison between the two approaches. The global silhouette values (GS) and the Davies-Bouldin index (DB) values are calculated for the selected solutions and comparisons are made based on the obtained values. GS values are calculated using Eq. 4, Eq. 5 and Eq. 6. The original DB index is defined for metric data in [9]. It has been modified in [31] for being used on graphs which is more suitable to be used in this study. The DB index is calculated using Eq. 8, Eq. 9, Eq. 10, Eq. 11.

$$\Delta(C_i) = \frac{1}{|C_i| * (|C_i| - 1)} \sum_{v_i, v_j \epsilon C_i, v_i \neq v_j} d(v_i, v_j) \tag{8}$$

where $\Delta(C_i)$ is the average diameter of cluster C_i, $|C_i|$ denotes the number of vertices in cluster C_i and $d(v_i, v_j)$ is the dissimilarity between the two vertices.

$$\delta(C_i, C_j) = \frac{1}{|C_i| * (|C_j|} \sum_{v_i \epsilon C_i, v_j \epsilon C_j} d(v_i, v_j) \tag{9}$$

where $\delta(C_i, C_j)$ is the average linkage between the two clusters.

$$DB_j(C_j) = max_{i \neq j}\{\frac{\Delta(C_i) + \Delta(C_j)}{\delta(C_i, C_j)}\} \quad (10)$$

where DB_j is the average similarity between cluster C_j and its most similar one.

$$DB(C) = \frac{1}{k}\sum_{j=1}^{k} DB_j(C_j) \quad (11)$$

where $DB(C)$ gives the DB index value of the clustering solution C. A lower value of DB index indicates a good clustering solution.

5. EXPERIMENTAL RESULTS

The results of the boolean function F based on the binary ϵ-indicator are given in Table 3 and Table 4. The first two rows are the contestant variations and the last row shows which variation is better than the other according to the ϵ-indicator. The "-" sign indicates that neither one of the variations has a precedence over the other. For test set FILES, all the variations listed in Table 1 are tested. For the CNET test set, the variations denoted as A1, A2, C1, C2, E1, E2, F1 and F2 in Table 1 are tested.

Table 3: Results for Test Set CNET

Var. 1	Var.2	Better Var.
A1	A2	A1
C2	A2	C2
C2	A1	C2
C1	A1	C1
C1	C2	C1
F1	E1	F1
F2	E2	F2

According to the obtained binary ϵ-indicator results for CNET, the C1 variation has produced the best Pareto approximation among the approaches using the objectives MMC and GS. It is possible to rank the algorithms as follows: C1, C2, A1, A2. Among the algorithms with objective functions OD and CO, E1 and E2 has the worst Pareto approximation sets. If we look at clustering solutions of these algorithms, we see that almost all graphs are partitioned into *maxCluster* number of clusters. Namely, just changing the objective functions of the same algorithm affects the performance of the algorithm badly. E1 and E2 are both dominated by F1 and F2. However it is not possible to distinguish F1 and F2 from each other.

For the test set FILES, it is also possible to rank the variations with objective functions MMC and GS based on the obtained binary ϵ-indicator values as C1, C2, A2, A1. Again the initialization and operators of MOCK are seen to be more successful than the random initialization method and the GraSC operators. To see the individual effect of the initialization based on MST, the algorithms are run with MST-initialization and GraSC operators (variations B1 and B2) and also with random initialization and MOCK-am operators (variations D1 and D2) on this test set. It is seen that B1/B2 has a better Pareto approximation than A1/A2 and C1/C2 are also better than D1/D2. In conclusion, initialization based on MST improves the performance of the variations.

Table 4: Results for Test Set FILES

Var.1	Var.2	Better Var.
B1	A1	B1
A2	A1	A2
A2	B2	A2
A2	C2	A2
A2	C2	-
B1	C1	-
B1	B2	-
C1	A1	C1
C1	B1	-
C1	C2	C1
C1	D1	C1
D1	A1	D1
F1	E1	F1
F2	E2	-

The approximation sets for the variation C1 which is the best among the all the variations using the objectives MMC and GS and F2 which is the best among the all the variations using the objectives OD and CO are given as a set of plots as Figure 1 and Figure 2 for CNET data set and as Figure 3 and Figure 4 for FILES data set. In Figure 1 and Figure 3 the x-axis gives the normalized GS value and the y-axis gives the converted MMC value. In Figure 2 and Figure 4 the x-axis gives the normalized CO value and the y-axis gives the normalized OD value. Note that the GS and the converted MMC are required to be maximized and OD and CO are required to be minimized. The most promising solutions, manually selected by the web mining expert are marked on the corresponding plots with an "X". The GS and DB indices of the actual clustering solutions corresponding to these points are listed in Table 5. As can be seen from the GS and DB index values, the C1 variation (mock-defaults, MMC and GS objectives and SPEA2) is better than the F2 variation (mock-defaults, OD and CO objectives and PESA-II) for both data sets.

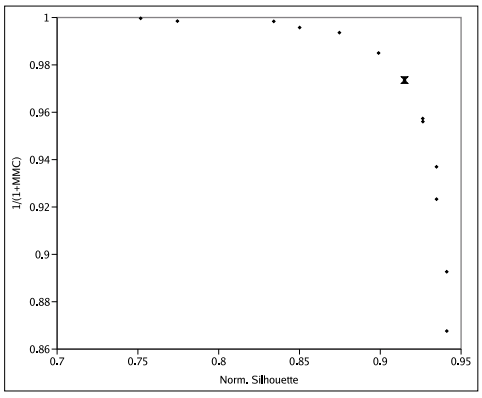

Figure 1: Pareto Appr. Plot of C1(CNET)

6. CONCLUSION

The experiments performed in this study are part of an ongoing research project on fast and efficient web recom-

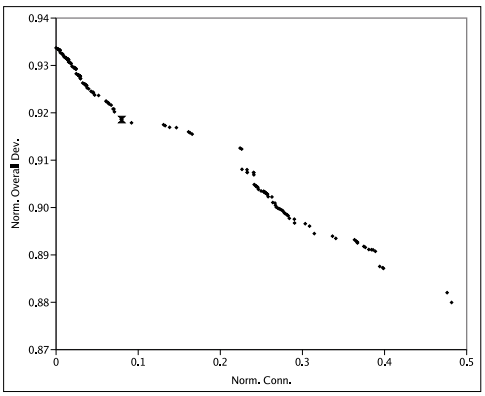

Figure 2: Pareto Appr. Plot of F2(CNET)

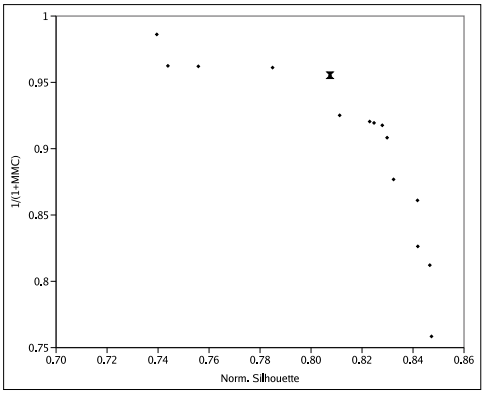

Figure 3: Pareto Appr. Plot of C1(FILES)

mendation. In web recommender systems, clustering can be applied offline to extract usage patterns. In such web recommenders, a successful recommendation highly depends on the quality of this clustering solution. In these types of applications, data to be clustered is in the form of user sessions which are sequences of web pages visited by the user. Sequence clustering is one of the important tools to work with this type of data. The main focus of this paper is to determine an effective MOEA to cluster sequence data given through pairwise similarity or dissimilarity data which can be represented as a graph. Two major MOEA approaches from literature for clustering such data sets are compared and the most suitable method is determined through the experiments. It can be seen from the results that an initialization based on a MST improves the clustering solution. Also optimization objectives exploiting properties of graphs are better than those which do not. When combined with these objectives and the MST based initialization technique, the SPEA2 algorithm performs better than PESA-II. As a future work, this clustering module will be integrated into a web recommender system.

7. ACKNOWLEDGMENTS

The authors were supported by the Scientific and Technological Research Council of Turkey (TUBITAK) EEEAG project 105E162.

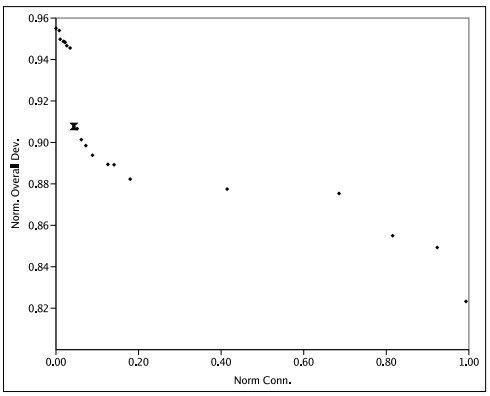

Figure 4: Pareto Appr. Plot of F2(FILES)

Table 5: Evaluation Results

	C1			F2		
Data Set	GS	DB	k	GS	DB	k
CNET	0.7979	0.7372	5	0.5247	1.2962	7
FILES	0.614	1.09757	3	0.048	1.9065	2

8. REFERENCES

[1] Clarknet www server log. http://ita.ee.lbl.gov/html/contrib/ClarkNet-HTTP.html.

[2] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[3] C. C.-K. and W. Y. A. An improved two-way partitioning algorithm with stable performance. *IEEE Trans. on Computed Aided Design*, 10:1502–1511, 1991.

[4] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, USA, 2002.

[5] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. Pesa-2: Region based selection in evolutionary multiobjective optimization. In *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, 2001.

[6] Ş. Gündüz and M. T. Özsu. A web page prediction model based on click-stream tree representation of user behavior. In *Proceedings of Ninth ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, August 2003.

[7] Ş. Gündüz Öğüdücü and A. Ş. Uyar. A graph based clustering method using a hybrid evolutionary algorithm. *WSEAS Transactions on Mathematics*, 3(3):731–736, 2004.

[8] A. Ş. Uyar and Ş. Gündüz Öğüdücü. A new graph-based evolutionary approach to sequence clustering. In *Proc. of 4th International Conference of Machine Learning and Applications*, 2005.

[9] D. Davies and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 1(2):224–227, 1979.

[10] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[11] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proc. of the 2001 IEEE Int. Conf. on Data Mining*, 2001.

[12] Z. H. G. M. S. H. Ding C, He X. A min-max cut algorithm for graph partitioning and data clustering. In *Proc. of the 2001 IEEE Int. Conf. on Data Mining*, pages 107–114, 2001.

[13] H. E. and S. R. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76:175–181, 2000.

[14] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. E. Arnold, London, UK, 2001.

[15] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. Morgan Kaufmann, 1993.

[16] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[17] J. Handl and J. Knowles. Supporting material. http://dbk.ch.umist.ac.uk/handl/mock/.

[18] J. Handl and J. Knowles. Evolutionary multiobjective clustering. In *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *LNCS*, pages 1081–1091. Springer, 2004.

[19] J. Handl and J. Knowles. Multiobjective clustering with automatic determination of the number of clusters. Technical report, UMIST TR-COMPSYSBIO-2004-02, Manchester, UK, 2004.

[20] J. Handl and J. Knowles. Evolutionary multiobjective clustering. In *Exploiting the trade-off – the benefits of multiple objectives in data clustering*, volume 3410 of *LNCS*, pages 547–560. Springer, 2005.

[21] J. Handl and J. Knowles. Evolutionary multiobjective clustering. In *Improving the scalability of multiobjective clustering*, pages 2371–2379. IEEE Press, 2005.

[22] J. Handl and J. Knowles. Multiobjective clustering around medoids. In *Proceedings of the Congress on Evolutionary Computation (CEC-2005)*, 2005.

[23] J. Handl and J. Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76, 2007.

[24] S. J. and M. J. Normalized cuts and image segmentation. *IEEE Trans. on Patterns Analysis and Machine Intelligence (PAMI)*, 2000.

[25] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[26] J. S. K. Charter and D. Szafron. Sequence alignment using fastlsa. In *International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, 2000.

[27] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley Sons, Inc., New York, USA, 1990.

[28] S. Kim. *Computational Biology and Genome Informatics*. World Scientific, 2003. Chapter 4.

[29] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154, 2001.

[30] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.

[31] N. Speer, C. Spieth, and A. Zell. Biological cluster validity indices based on the gene ontology. In A. F. F. et al., editor, *Advances in Intelligent Data Anylsis VI: 6th International Symposium on Intelligent Data Analysis (IDA 2005)*, volume 3646 of *Lecture Notes in Computer Science (LNCS)*, pages 429–439. Springer, 2005.

[32] E. Zitzler, M., and L. Thiele. Spea2: Improving the strength pareto evolutonary algorithm for multiobjective optimization. Technical report, Swiss Federal Institute of Technology, 2002.

[33] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. Fonseca. Performance assessment of multiobjective optimizers: an analysis and review, 2002.